

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-309285

(43)Date of publication of application : 04.11.1994

(51)Int.Cl.

G06F 15/16

G06F 13/00

(21)Application number : 05-099901

(71)Applicant : FUJITSU LTD

(22)Date of filing : 27.04.1993

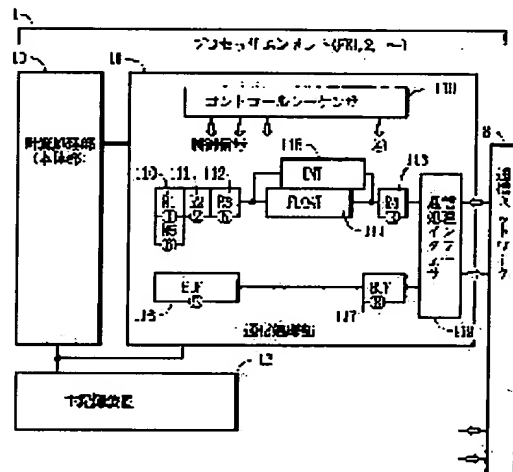
(72)Inventor : ICHIKAWA SHINICHI

(54) COMMUNICATION PROCESSING CIRCUIT FOR PARALLEL COMPUTER

(57)Abstract:

PURPOSE: To reduce the overhead of a processor element accompanied with an inter-processor element communication processing by providing a specific communication processing part, and selectively executing a prescribed communication processing independently of an operation at a calculation processing part.

CONSTITUTION: This circuit is constituted of a calculation processing part 10 and a communication processing part 11. Then, the communication processing part 11 is equipped with registers 110-113 which store logic address (1) of a processor element 1, number (2) of entire processors, logical data (5) and (6), and communication data (3) communicated from a communication network 3, buffers 116-117, each kind of arithmetic unit 114 and 115, control sequencer 118 which operates communication control, and communication network interface 119, as a hardware mechanism used exclusively for a global processing. Then, after the circuit is activated by a software instruction executed by the calculation processing part 10, the global communication processing is executed by the hardware mechanism independently of the calculation processing part 10.



LEGAL STATUS

[Date of request for examination]

24.04.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japanese Patent Office

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-309285

(43)公開日 平成6年(1994)11月4日

(51)Int.Cl.⁵

G 0 6 F 15/16
13/00

識別記号

3 9 0 Z 9190-5L
3 5 3 C 7368-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数 2 O L (全 8 頁)

(21)出願番号 特願平5-99901

(22)出願日 平成5年(1993)4月27日

(71)出願人 000005223

富士通株式会社
神奈川県川崎市中原区上小田中1015番地

(72)発明者 市川 真一

神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(74)代理人 弁理士 井桁 貞一

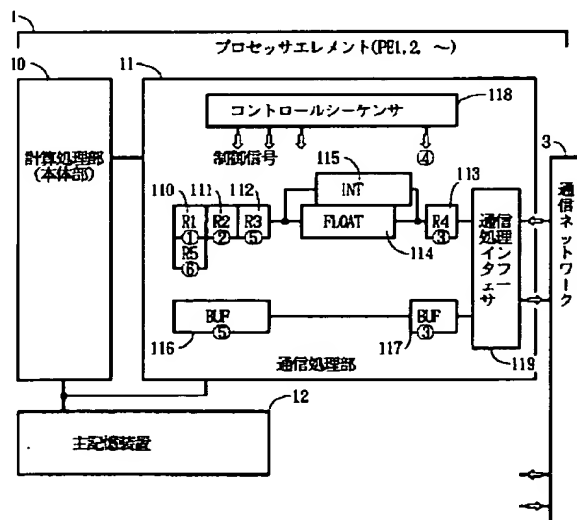
(54)【発明の名称】 並列計算機における通信処理回路

(57)【要約】

【目的】 本発明は、並列計算機における通信処理回路に関し、プロセッサエレメント間の通信処理に伴うプロセッサエレメントでのオーバーヘッドを削減する。

【構成】 プロセッサエレメント内に、計算処理部とは別に、プロセッサエレメントの論理アドレス①と、全プロセッサエレメントの数②と、通信データ③とを格納するレジスタ、バッファと、各種の演算器と、通信制御を行うコントロールシーケンサとからなる通信処理部を設け、計算処理部で実行されるソフトウェアからの指示で、上記レジスタにプロセッサエレメントの論理アドレス①、プロセッサエレメントの数②を設定した後、該設定された論理アドレス①と、プロセッサエレメント数②と、第何回目の通信であるかを指示しているコントロールシーケンサのシーケンス番号④とで定まるバイナリツリー方法による大域的な通信処理（データの送信、データ受信、演算、又は、演算結果の送信）を、計算処理部での動作とは独立に実行するように構成する。

本発明の一実施例を模式的に示した図



【特許請求の範囲】

【請求項1】分散型メモリ(12)を備えた複数のプロセッサエレメント(1)が、通信ネットワーク(3)を介して接続されている並列計算機におけるプロセッサエレメント(1)での通信処理回路であって、各プロセッサエレメント(1)内に、計算処理部(10)とは別に、プロセッサエレメントの論理アドレス(①)と、全プロセッサエレメントの数(②)と、計算処理部(10)からのローカルデータ(⑤、⑥)と、通信ネットワーク(3)から受信した通信データ(③)とを格納するレジスタ(110,111,112,120,113)、バッファ(116,117)と、各種の演算器(114,115)と、通信制御を行うコントロールシーケンサ(118)と、通信ネットワークインタフェース(119)とからなる通信処理部(11)を設け、計算処理部(10)で実行されるソフトウェアからの指示で、上記レジスタ(110,111)にプロセッサエレメントの論理アドレス(①)、プロセッサエレメントの数(②)を設定した後、該設定された論理アドレス(①)と、プロセッサエレメント数(②)と、第何回目の通信であるかを指示しているコントロールシーケンサ(118)のシーケンス番号(④)とで定まる通信処理(データの送信、又は、データ受信、演算、又は、演算結果の送信)を、選択的に、上記計算処理部(10)での動作とは独立に実行することを特徴とする並列計算機における通信処理回路。

【請求項2】上記通信処理として、バイナリツリー手順により、大域的な演算を行うことを特徴とする請求項1に記載の並列計算機における通信処理回路。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、分散型メモリを備えた並列計算機における通信処理方法に関し、特に、並列計算機を構成している複数の全プロセッサエレメント(PE1, ~)上でデータを参照して行う大域的な処理を行う際の通信処理回路に関する。

【0002】あらゆる工業分野、技術開発分野で設計を行う際には、偏微分方程式を解いたり、構造を解析したりして、実験によらずに数値シミュレーションにより、製品の特性、性能を予測することが重要となってきた。しかも、この数値シミュレーションには、年々より高速のコンピュータが必要となっており、中央処理装置(CPU)を数多く連ねた並列コンピュータは、計算能力に対する増大する要求に答える手段として、その利用が検討されはじめている。

【0003】分散メモリ型並列計算機上で、このような数値シミュレーションの並列処理を行う時には、ホストからデータ用の配列を、各プロセッサエレメント(PE1, PE2, ~)に分割配置してデータの更新を行う。

【0004】このような並列処理に必要な通信処理は、数値モデル上で相互作用の及ぶ範囲のデータを持つプロ

セッサエレメント(PE1, PE2, ~)から参照すべきデータのコピーを受け取る局所的な処理と、全プロセッサエレメント(PE1, PE2, ~)上のデータを参照して演算を行う、大域的な処理とがある。

【0005】このような、大域的な処理においては、プロセッサエレメント(PE1, PE2, ~)の数が多くなると、処理時間も大きくなるが、該処理時間に含まれる通信処理時間も無視できなくなること、又、該大域的な処理においては、他のプロセッサエレメントからデータをもたらって、所定の演算をした後、別のプロセッサエレメントに送信するといった処理であるにも関わらず、データを受信する毎に、該プロセッサエレメント内のソフトウェア(即ち、アプリケーション)に割り込みが発生して、該アプリケーションが擾乱され、該並列計算機全体としての処理能力が低下することから、上記プロセッサエレメント内で実行されているアプリケーション等に対する影響の少ない通信処理回路が要求される。

【0006】

【従来の技術】図3は、並列計算機での従来の通信処理方法を説明する図であり、図3(a)はメモリ分散型並列計算機の構成例を示し、図3(b)は、プロセッサエレメント間で送受信されるデータのフォーマット例を示している。

【0007】先ず、図3(b)に示したデータフォーマットにおいて、先頭のヘッダ部は、通信先のプロセッサエレメント(PE1, ~)1のアドレス(SA)と、該通信先のプロセッサエレメント(PE1, ~)1に対する割り込み(INT)の有無と、データのクラス(アプリケーションが優先して処理する必要のあるデータか否かを識別するクラス)等の制御情報で構成されている。

【0008】送信側のプロセッサエレメント(PEn)1のアプリケーション、例えば、通信ライブラリが、プロセッサエレメント(PE1)1にデータの送信を行う場合、所定のデータを、主記憶装置12上に用意した後、所定の条件を指示して、通信処理部11内の、例えば、ダイレクトメモリアクセス機構(DMA)110を起動する。

【0009】該ダイレクトメモリアクセス機構(DMA)110は、指示された条件の元で、プロセッサエレメント(PEn)1の主記憶装置12の所定のアドレスから、所定のデータ長のデータを読み取り、通信ネットワーク3を介して、プロセッサエレメント(PE1)1にデータを転送する。

【0010】プロセッサエレメント(PE1)1の通信処理部11では、送信されてきた通信データ③のヘッダ部を参照して、自己のプロセッサエレメント(PE1)1に対する送信データであって、且つ、割り込みフラグ(INT)が“1”になっていると、本体部(計算処理部)10で実行されているオペレーションシステム(OS)に割り込みを発生する。

【0011】本体部10で実行されるオペレーションシ

ステム(OS)では、上記割り込みを受け付けると、通信データ③の、上記ヘッダ部を取り込み、通信先のチェックをした後、データ長等を参照して、ダイレクトメモリアクセス機構(DMA) 110 を起動する。

【0012】起動されたダイレクトメモリアクセス機構(DMA) 110 では、指示されたデータ転送条件の元で、通信ネットワーク 3から通信データ③のデータ部を読み取り、主記憶装置 12 に転送する。

【0013】このように、従来の通信処理では、所定のプロセッサエレメント(PE_i) 1 からのデータ転送がある毎に、通信先のプロセッサエレメント(PE_j) 1 の本体部 10で実行されているオペレーションシステム(OS)に割り込まれ、少なくとも、ヘッダ部を読み取る為に、通信データ③を通信処理部 11 から本体部 10 まで転送する動作が実行される。

【0014】

【発明が解決しようとする課題】従って、現在の並列計算機では、主記憶装置 12 に対するメモリアクセスに比べて、通信処理のスループットが格段に遅いため、並列処理の効果を得るためには、出来るだけ通信処理、及び、通信に関連する処理の時間を短くする工夫が必要である。

【0015】前記数値シミュレーションを並列処理する際に必要な、通信を要する大域的処理の例は、次のようなものである。

a) 最大最小値探索(大小比較)
b) 大域的論理演算(論理和、排他的論理和等)
c) 総和計算(浮動小数点、整数の加算)
d) 処理要素に分割された全てのデータの共有(合同)

これらの複合的な通信処理を、従来の計算処理部 10 と通信処理部 11 とから成り立つプロセッサエレメント(PE_i) 1 で、上記の如きメッセージパッシング(メッセージ受け渡し)機構、例えば、ダイレクトメモリアクセス機構(DMA) 110、割り込み機構等により行う際には、通信そのものよりも、通信を起動するソフトウェアや、通信データ③、例えば、ヘッダ部、の受け取り、送込に関わるプロセッサエレメント(PE_i) 1 内の処理に時間を多く費やされる。

【0016】上記の如き、大域的処理では、受け取ったデータに一つの演算を施した後、直ちに、再び、通信ネットワーク 3へ送出する決まった処理であるにもかかわらず、メッセージパッシング(データの受け渡し)による一般的な通信機構を用いることは、通信ソフトウェア(上記通信ライブラリ)内の通信先チェックなどの処理の重複や、本体部 10 と通信処理部 11 との間のデータの移動などのオーバーヘッドを被りやすい。しかも、これらの大域的処理で、最も有効な、バイナリツリーアルゴリズム(後述の図2参照)では、演算処理が必要なのは、全てのプロセッサエレメント(PE_i) 1 ではなく、一部のプロセッサエレメントである。このため、これらの

処理を頻繁に行う数値シミュレーションでは、演算処理を行わない他のプロセッサエレメント(PE_j)に待ちが生じて稼働率が下がり、並列処理の効果が出にくい。

【0017】本発明は上記従来の欠点に鑑み、並列計算機で行われる数値シミュレーション等の並列処理に必要な、数値シミュレーションの並列処理効果を下げる、複合通信処理を、高速に行うことができる通信処理の方法を提供することを目的とするものである。

【0018】

【課題を解決するための手段】図1は、本発明の一実施例を模式的に示した図であり、図2はバイナリツリーによる大域的処理を説明する図である。上記の問題点は下記の如くに構成した並列計算機における通信処理方法によって解決される。

【0019】(1) 分散型メモリ(主記憶装置)12を備えた複数のプロセッサエレメント1が、通信ネットワーク3を介して接続されている並列計算機におけるプロセッサエレメント1での通信処理回路であって、各プロセッサエレメント1内に、計算処理部10とは別に、プロセッサエレメントの論理アドレス①と、全プロセッサエレメントの数②と、計算処理部10からローカルデータ⑤、⑥と、通信ネットワーク3から受信した通信データ③とを格納するレジスタ110,111,112,120,113,バッファ116,117と、各種の演算器114,115と、通信制御を行うコントロールシーケンサ118と、通信ネットワークインタフェース119とからなる通信処理部11を設け、計算処理部10で実行されるソフトウェアからの指示で、上記レジスタ110,111にプロセッサエレメントの論理アドレス①、プロセッサエレメントの数②を設定した後、該設定された論理アドレス①と、プロセッサエレメント数②と、第何回目の通信であるかを指示しているコントロールシーケンサ118のシーケンス番号④とで定まる通信処理(データの送信、又は、データ受信、演算、又は、演算結果の送信)を、選択的に、上記計算処理部10での動作とは独立に実行するように構成する。

【0020】(2) 上記通信処理として、バイナリツリー手順により、大域的な演算を行うように構成する。

【0021】

【作用】前述のように、分散メモリ型並列計算機で、例えば、数値シミュレーションを並列処理する際に必要な、プロセッサエレメント(PE_i)間の通信を必要とする大域的な処理の例として、a) 最大最小値探索(大小比較)、b) 大域的論理演算(論理和、排他的論理和等)、c) 総和計算(浮動小数点、整数の加算)、d) データ列の合同(繋ぎ合わせ)等があるが、この大域的処理で最も有効な通信手段として、図2に示したバイナリツリーアルゴリズムが知られている。

【0022】図2から明らかなように、バイナリツリーによる通信処理では、該並列計算機を構成しているプロセッサエレメント(PE1, PE2, ...)の数によって、該バ

イナリーツリーの構成が決められ、図2の構成例では、例えば、奇数番号のプロセッサエレメント(PE1, PE3, PE5, ...) 1では、通信ネットワーク3から通信データ③を受信して、予め、定められている演算処理(OPRで示す)を実行するか、更に、実行した演算結果を、1つ、又は、2つ、又は、4つ若番のプロセッサエレメント(PEi) 1に転送するかに定形化されており、どの通信処理を行うかは、上記バイナリーツリーの第何番目の通信であるかによって決まっている。

【0023】例えば、プロセッサエレメント(PE1) 1は、データを受信して、所定の演算を繰り返すのみであるが、プロセッサエレメント(PE3, PE7, ...) 1では、第1回目の通信処理で、演算処理を行い、演算結果を他のプロセッサエレメント(PE1) 1に送信するのみであり、プロセッサエレメント(PE5, ...) 1では、第1回目の通信処理では演算処理のみであり、第2回目の通信処理では、演算した結果を他のプロセッサエレメント(PE1) 1に送信するといったように、プロセッサエレメント(PEi) のプロセッサエレメントアドレス(番号) ①と、第何回目の通信処理(これは、コントロールシーケンサ番号④で決まる)であるかにより、通信処理の内容が定形化されている。

【0024】又、偶数番号のプロセッサエレメント(PE2, PE4, ...) 1では、自己の持っているデータ(即ち、ホストから配分されているデータ、ローカルデータ) ⑤を他のプロセッサエレメント(PE1, PE3, ...) 1に送信するのみである。

【0025】本発明は、この点に着目して、各プロセッサエレメント1内に、本体部である計算処理部10とは別に、プロセッサエレメントの論理アドレス①と、全プロセッサエレメントの数②と、自己の持っているローカルデータ⑤、⑥と、通信ネットワーク3から受信した通信データ③とを格納するレジスタ110, 111, 112, 120, 113, バッファ116, 117と、各種の演算器114, 115と、通信制御を行うコントロールシーケンサ118と、通信ネットワークインタフェース(119)とからなる通信処理部11を設け、該計算処理部10で実行されるソフトウェアからの指示で、上記レジスタ110, 111にプロセッサエレメントの論理アドレス①、プロセッサエレメントの数②を設定した後、該設定された論理アドレス①と、プロセッサエレメント数②と、第何回目の通信であるかを指示しているコントロールシーケンサ118のシーケンス番号④とで定まる通信処理(データの送信、又は、データ受信、演算、又は、演算結果の送信)を、選択的に、上記計算処理部10での動作とは独立に実行するように構成したものである。

【0026】従って、従来のように、汎用的なメッセージパッシング(メッセージの受け渡し)の通信ソフトウェアを多数回実行することによるオーバーヘッドを少なくでき、又、定形的な処理となるバイナリーツリーアルゴ

リズムを、簡単なハードウェア機構で実行することにより、プロセッサエレメント(PEi)の本体部である計算処理部でのメモリアクセス、入出力処理と競合することなく、演算処理を実行でき、又、通信ネットワークの通信データ③を、各プロセッサエレメント(PEi)の計算処理部へ移動させずに済む為、通信処理を高速化できる。この結果、複合通信処理の時間を短縮することができ、全プロセッサエレメント(PEi)の稼働率を向上させることができる効果が得られる。

【0027】

【実施例】以下本発明の実施例を図面によって詳述する。前述の図1は、本発明の一実施例を模式的に示した図であり、図2は、バイナリーツリーによる大域的処理を説明する図である。

【0028】本発明においては、各プロセッサエレメント1内に、本体部である計算処理部10とは別に、プロセッサエレメントの論理アドレス①と、全プロセッサエレメントの数②と、ローカルデータ⑤、⑥と、通信ネットワークから受信した通信データ③とを格納するレジスタ110, 111, 112, 120, 113, バッファ116, 117と、各種の演算器114, 115と、通信制御を行うコントロールシーケンサ118と、通信ネットワークインタフェース(119)とからなる通信処理部11を設け、該計算処理部10で実行されるソフトウェアからの指示で、上記レジスタ110, 111にプロセッサエレメントの論理アドレス①、プロセッサエレメントの数②を設定した後、該設定された論理アドレス①と、プロセッサエレメント数②と、第何回目の通信であるかを指示しているコントロールシーケンサ118のシーケンス番号④とで定まる通信処理(データの送信、又は、データ受信、演算、又は、演算結果の送信)を、選択的に、上記計算処理部10での動作とは独立に実行する手段が、本発明を実施するのに必要な手段である。尚、全図を通して同じ符号は同じ対象物を示している。

【0029】以下、図1、図2によって、本発明の分散メモリ型並列計算機における通信処理回路の構成と動作を説明する。分散メモリ型並列計算機のプロセッサエレメント(PE1, PE2, ...) 1は、図1に示されているように、計算処理部10と通信処理部11とから成り、通信処理部11に、大域的処理を行う専用のハードウェア機構として、プロセッサエレメントの論理アドレス①と、全プロセッサエレメントの数②と、ローカルデータ⑤、⑥と、通信ネットワーク3から受信した通信データ③とを格納するレジスタ110, 111, 112, 120, 113, バッファ116, 117と、各種の演算器114, 115と、通信制御を行うコントロールシーケンサ118と、通信ネットワークインタフェース119を設け、計算処理部10で実行されるソフトウェア命令により起動された後は、全てハードウェアにより、計算処理部10とは独立に、大域的な通信処理が、以下に説明するハードウェア機構1、2で実行され

る。

【0030】「ハードウェア機構1」：バイナリツリーによる通信先の決定と、送受信を制御する機構として、プロセッサエレメントの論理アドレス①を設定するレジスタ(R1) 110と、バイナリツリーによる通信処理の定形化に關する全プロセッサエレメントの数②を設定するレジスタ(R2) 111と、主記憶装置 12 上のデータ、即ち、ローカルデータ⑤を格納しておくレジスタ(R3) 112、及び、バッファ(BUF) 116 と、ローカルデータのデータ長⑥を格納しておくレジスタ(R5) 120と、通信ネットワーク 3からの通信データ③を格納するレジスタ(R3) 113、及び、バッファ(BUF) 117 と、現在の通信処理が、上記バイナリツリーによる通信処理における第何回目の通信処理であるか、即ち、シーケンス番号④を指示して、所定の制御信号を出力するコントロールシーケンサ 118とを通信処理部 11 内に設ける。

【0031】コントロールシーケンサ 118は、レジスタ(R1) 110にあるプロセッサエレメントアドレス①、及び、レジスタ(R2) 111に設定されている全プロセッサエレメントの数②をもとに通信処理を制御する。

【0032】バイナリツリーによる通信先の決定方法を図2示す。図2から明らかなように、バイナリツリーによる通信処理では、例えば、奇数番号のプロセッサエレメント(PE1, PE3, PE5, ...) 1 では、通信ネットワーク 3から通信データ③を受信して、予め、定められている演算処理(OPRで示す)を実行するか、更に、実行した演算結果を、1つ、又は、2つ、又は、4つ番番のプロセッサエレメント(PEj) 1 に転送するかに定形化されており、どの通信処理を行うかは、上記バイナリツリーの第何番目の通信であるか、即ち、上記シーケンス番号④によって決まっている。

【0033】例えば、プロセッサエレメント(PE1) 1 は、データを受信して、所定の演算を繰り返すのみであるが、プロセッサエレメント(PE3, PE7, ...) 1 では、第1回目の通信処理で、演算処理を行い、演算結果を他のプロセッサエレメント(PE1) 1に送信するのみであり、プロセッサエレメント(PE5, ...) 1 では、第1回目の通信処理では演算処理のみであり、第2回目の通信処理では、演算した結果を他のプロセッサエレメント(PE1) 1 に送信するといったように、プロセッサエレメント(PEi) のプロセッサエレメントアドレス(番号) ①と、第何回目の通信処理であるかを示すシーケンス番号④により、通信処理の内容が定形化されている。

【0034】又、偶数番号のプロセッサエレメント(PE2, PE4, ...) 1 では、自己の持っているデータ(即ち、ホストから配分されているローカルデータ) ⑤を他のプロセッサエレメント(PE1, PE3, ...) 1 に送信するのみである。

【0035】そこで、本発明においては、上記レジスタ(R1) 110に設定されている自己のプロセッサエレメント

アドレス(プロセッサエレメント番号) ①と、レジスタ(R2) 111に設定されている、バイナリツリーの全体の構成を決定する全プロセッサエレメントの数②と、該バイナリツリーによる通信処理において、第何回目の通信処理であるかを指示するコントロールシーケンサ 118のシーケンス番号④とによって、自己の処理する通信処理の形態を決定する。

【0036】「ハードウェア機構2」：上記ハードウェア機構1の制御により動作する浮動小数点加算器(FLOAT) 114、整数加算器(INT) 115、又は、演算処理時にアクセスされる、前述のレジスタ(R3) 112、レジスタ(R4) 113、レジスタ(R5) 120、及び、バッファ(BUF) 116、117を通信処理部 11に設ける。

【0037】上記レジスタ(R4) 113、及びバッファ(BUF) 117は、通信ネットワーク 3から直接通信データ③を受け取り、又、通信ネットワーク 3へ直接、ホストから配分されているローカルデータ⑤、或いは、自己の通信処理部 11内の上記浮動小数点加算器(FLOAT) 114、整数加算器(INT) 115での演算結果データを通信ネットワーク 3に送出させる。これらのハードウェア機構は、次の実施例のように動作して大域的処理を行う。

【0038】「段階1」：計算処理部 10で実行されるソフトウェア、例えば、前述の通信ライブラリの指示により、上記「ハードウェア機構1」が動作を開始する。この時、ソフトウェアからプロセッサエレメントの論理アドレス①を受け取り、上記レジスタ(R1) 110に設定する。又、ソフトウェアから大域的処理を行う対象であるローカルデータ⑤を、前述の大域的処理a)、b)、c)の場合は、ローカルデータ用のレジスタ(R3) 112に受け取り、大域的処理d)の場合は、バッファ(BUF) 116に受け取る。又、前述の大域的処理d)の場合は、ローカルデータの長さ⑤を、データ長用のレジスタ(R5) 120に受け取る。

【0039】「段階2」：演算を担当するプロセッサエレメント(PEi) 1は、バイナリツリーアルゴリズムに従って、他のプロセッサ演算(PEj) 1から通信データ③を受信し、次の処理を行う。

【0040】1) 大域的処理a)、b)、c) {大小比較、論理和、排他的論理和、論理演算等}の場合、レジスタ(R3) 112にあるローカルデータ⑤と、通信ネットワーク3から転送され、レジスタ(R4) 113に格納されている通信データ③との間で、それぞれの演算を行い、結果は、バイナリツリーアルゴリズムに従って、ローカルデータ用レジスタ(R3) 112に格納するか、又は、他のプロセッサエレメント(PEj) 1に送信する。

【0041】2) 大域的処理d) {合同、即ち、データの結合}の場合、ネットワーク 3から転送され、バッファ(BUF) 117に格納されている通信データ③を、上記データ長用のレジスタ(R5) 120を参照して、バッファ(BUF) 116にあるローカルデータ⑤の最後尾に追加して書

10

20

30

40

50

き込み、合計したデータ長を、上記データ長用のレジスタ(R5) 120Cに書き込む。

【0042】上記「段階1、2」での処理をバイナリツリーが収束するまで繰り返す。

「段階3」：バイナリツリーの頂点に立つプロセッサエレメント（例えば、図1に示したバイナリツリー構成では、PE1）1は、最終結果を他の全てのプロセッサエレメントに放送し、次の処理に備える。

【0043】このように、本発明においては、並列計算機における通信処理回路において、プロセッサエレメント内に、計算処理部とは別に、プロセッサエレメントの論理アドレス①と、全プロセッサエレメントの数②と、ローカルデータ⑤と、通信データ③とを格納するレジスタ、バッファと、各種の演算器と、通信制御を行うコントロールシーケンサとからなる通信処理部を設け、計算処理部で実行されるソフトウェアからの指示で、上記レジスタにプロセッサエレメントの論理アドレス①、プロセッサエレメントの数②を設定した後、該設定された論理アドレス①と、プロセッサエレメント数②と、第何回目の通信であるかを指示しているコントロールシーケンサのシーケンス番号④とで定まるバイナリツリー方法による大域的な通信処理（データの送信、又は、データ受信、演算、又は、演算結果の送信）を、選択的に、計算処理部（本体部）での動作とは独立に実行するように構成したところに特徴がある。

【0044】

【発明の効果】以上、詳細に説明したように、本発明の並列計算機における通信処理回路によれば、汎用的なメッセージパッシング（メッセージの受け渡し）の通信ソフトウェアを多数回実行することによるオーバーヘッドを避けることができ、又、定形的処理である、バイナリツリーアルゴリズムをハードウェア機構で実行することにより、プロセッサエレメント(PE1, PE2, ...)の他の入出力やメモリアクセスと競合せずに、演算処理の制御を*

*行わせることができ、また通信ネットワークからの通信データ③をプロセッサエレメント(PE1, PE2, ...)の計算処理部（本体部）へ移動せずに済むため、処理が高速化できる。この結果、複合通信処理の時間を短縮でき、全プロセッサエレメントの稼働率を向上させることができる効果がある。

【図面の簡単な説明】

【図1】本発明の一実施例を模式的に示した図

【図2】バイナリツリーによる大域的処理を説明する図

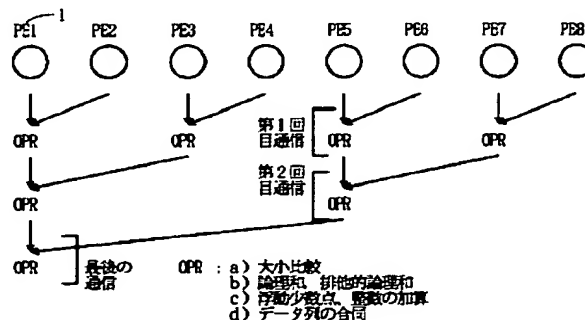
【図3】並列計算機での従来の通信処理方法を説明する図

【符号の説明】

- | | | | |
|-----|-------------------------------|----------|----------|
| 1 | プロセッサエレメント(PE1, PE2, ...) | 11 | 通信処理部 |
| 10 | 計算処理部（本体部） | 111 | レジスタ(R |
| 110 | レジスタ(R1) | 112 | レジスタ(R3) |
| 2) | | 113 | レジスタ(R |
| 112 | レジスタ(R3) | 4) | |
| 114 | 浮動小数点加算器(FLOAT) | 20 | |
| 115 | 整数加算器(INT) | 116, 117 | バッ |
| | | | ファ(BUF) |
| 118 | コントロールシーケンサ | | |
| 119 | 通信ネットワークインターフェース | | |
| 120 | レジスタ(R5) | | |
| 12 | 分散型メモリ（主記憶装置） | 3 | 通信ネット |
| | | | ワーク |
| ① | プロセッサエレメントアドレス（プロセッサエレメントの番号） | | |
| ② | プロセッサエレメントの数 | | |
| ③ | 通信データ | | |
| ④ | シーケンス番号 | | |
| ⑤ | ローカルデータ | | |
| ⑥ | ローカルデータ（データ長） | | |

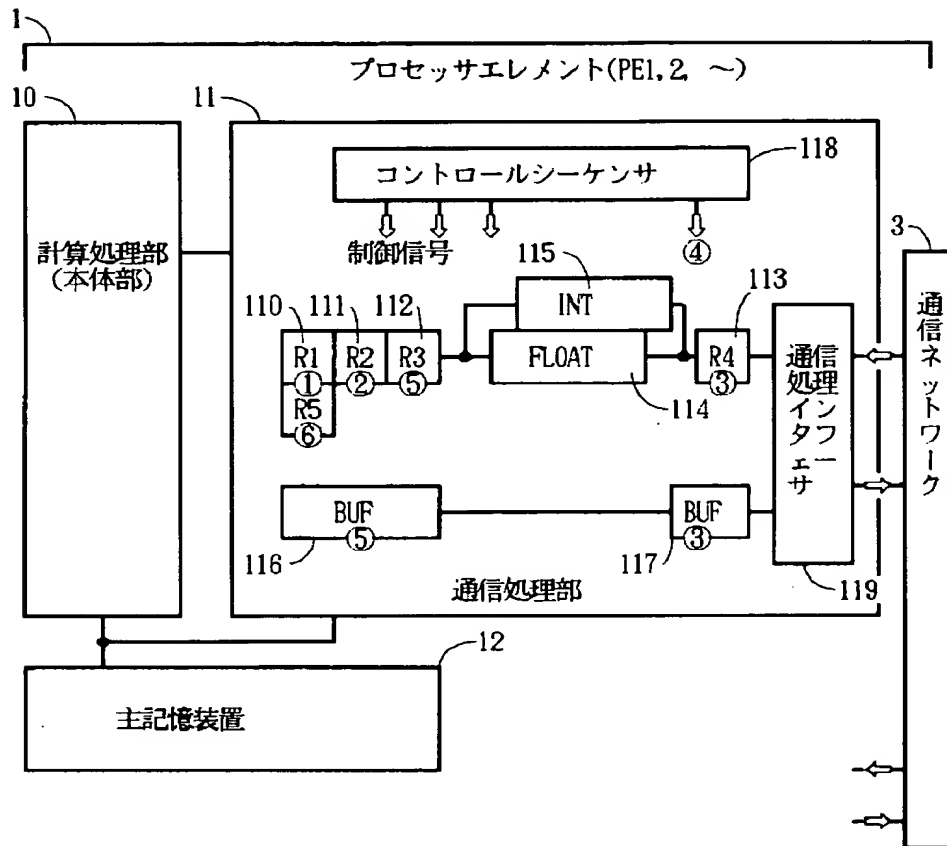
【図2】

バイナリツリーによる大域的処理を説明する図



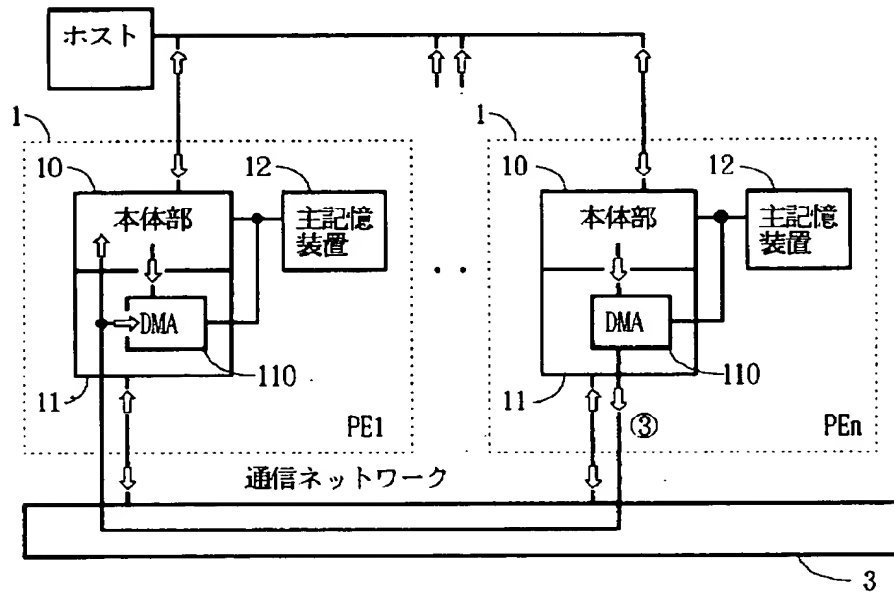
【図1】

本発明の一実施例を模式的に示した図

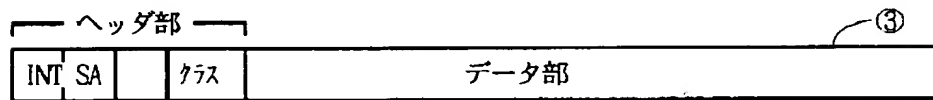


【図3】

並列計算機での従来の通信処理方法を説明する図



(a)



(b)